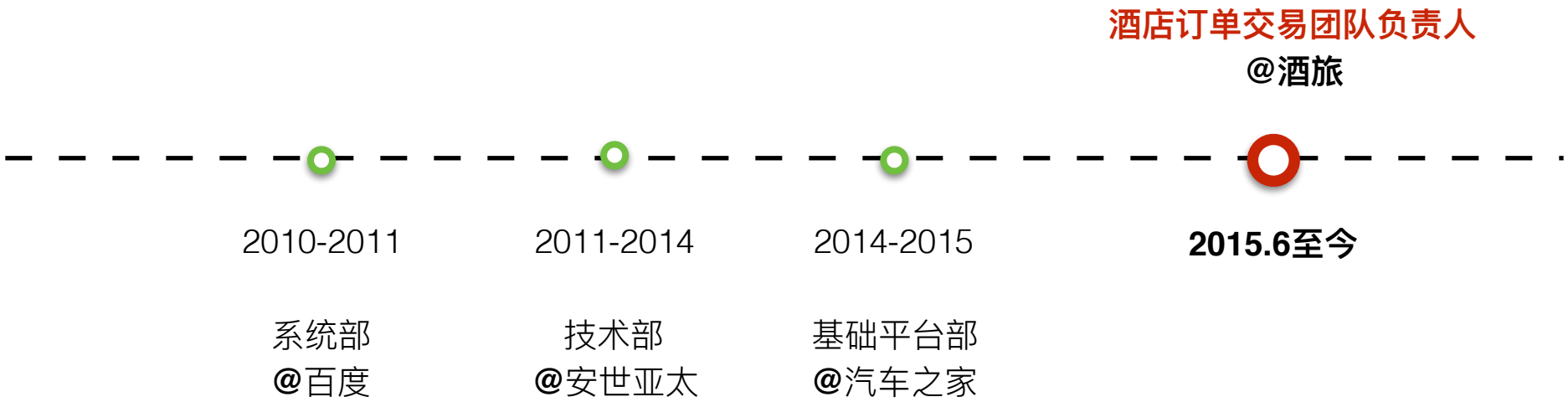
酒店订单系统架构实践

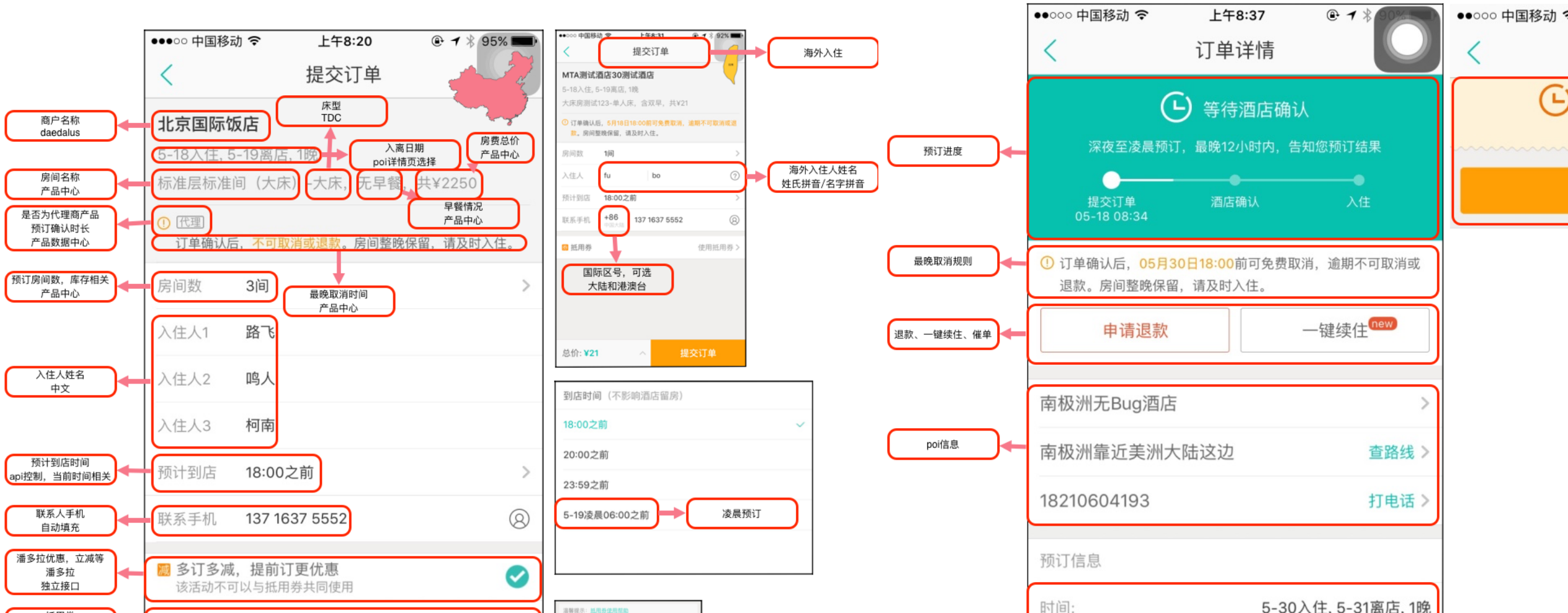
吴革@酒旅后台研发团队 201611

个人简介


酒店订单交易团队负责人
@酒旅
2010-2011
系统部
@百度
2011-2014
技术部
@安世亚太
2014-2015
基础平台部
@汽车之家
2015.6至今

目录

业务简介-功能介绍


提交订单
北京国际饭店
床型
TDC
5-18入住, 5-19离店, 1晚
入离日期
poi详情页选择
房费总价
产品中心
标准层标准间（大床）
大床, 无早餐, 共¥2250
早餐情况
产品中心
代理
订单确认后，不可取消或退款。房间整晚保留，请及时入住。
房间数 3间
最晚取消时间
产品中心
入住人1 路飞
入住人2 鸣人
入住人3 柯南
预计到店 18:00之前
联系手机 137 1637 5552
多订多减，提前订更优惠
该活动不可以与抵用券共同使用
商户名称
daedalus
房间名称
产品中心
是否为代理商产品
预订确认时长
产品数据中心
预订房间数，库存相关
产品中心
入住人姓名
中文
预计到店时间
api控制，当前时间相关
联系人手机
自动填充
潘多拉优惠，立减等
潘多拉
独立接口
提交订单
MTA测试酒店30测试酒店
海外入住
海外入住人姓名
姓氏拼音/名字拼音
国际区号，可选
大陆和港澳台
总价: ¥21
提交订单
到店时间（不影响酒店留房）
18:00之前
20:00之前
23:59之前
5-19凌晨06:00之前
凌晨预订
订单详情
等待酒店确认
深夜至凌晨预订，最晚12小时内，告知您预订结果
提交订单
05-18 08:34
酒店确认
入住
预订进度
最晚取消规则
订单确认后，05月30日18:00前可免费取消，逾期不可取消或退款。房间整晚保留，请及时入住。
退款、一键续住、催单
申请退款
一键续住
new
poi信息
南极洲无Bug酒店
南极洲靠近美洲大陆这边
查路线
18210604193
打电话
预订信息
时间:
5-30入住, 5-31离店, 1晚

业务简介-系统发展

2012年12月

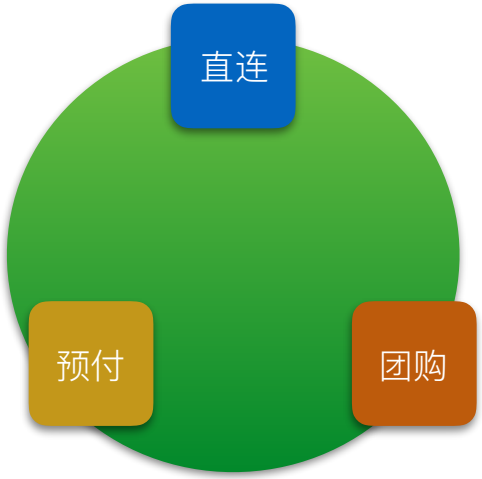
hotel-server

2015年3月

hotel-order

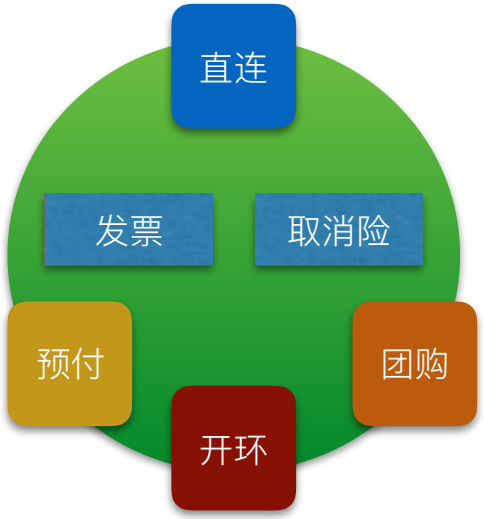
2015年10月

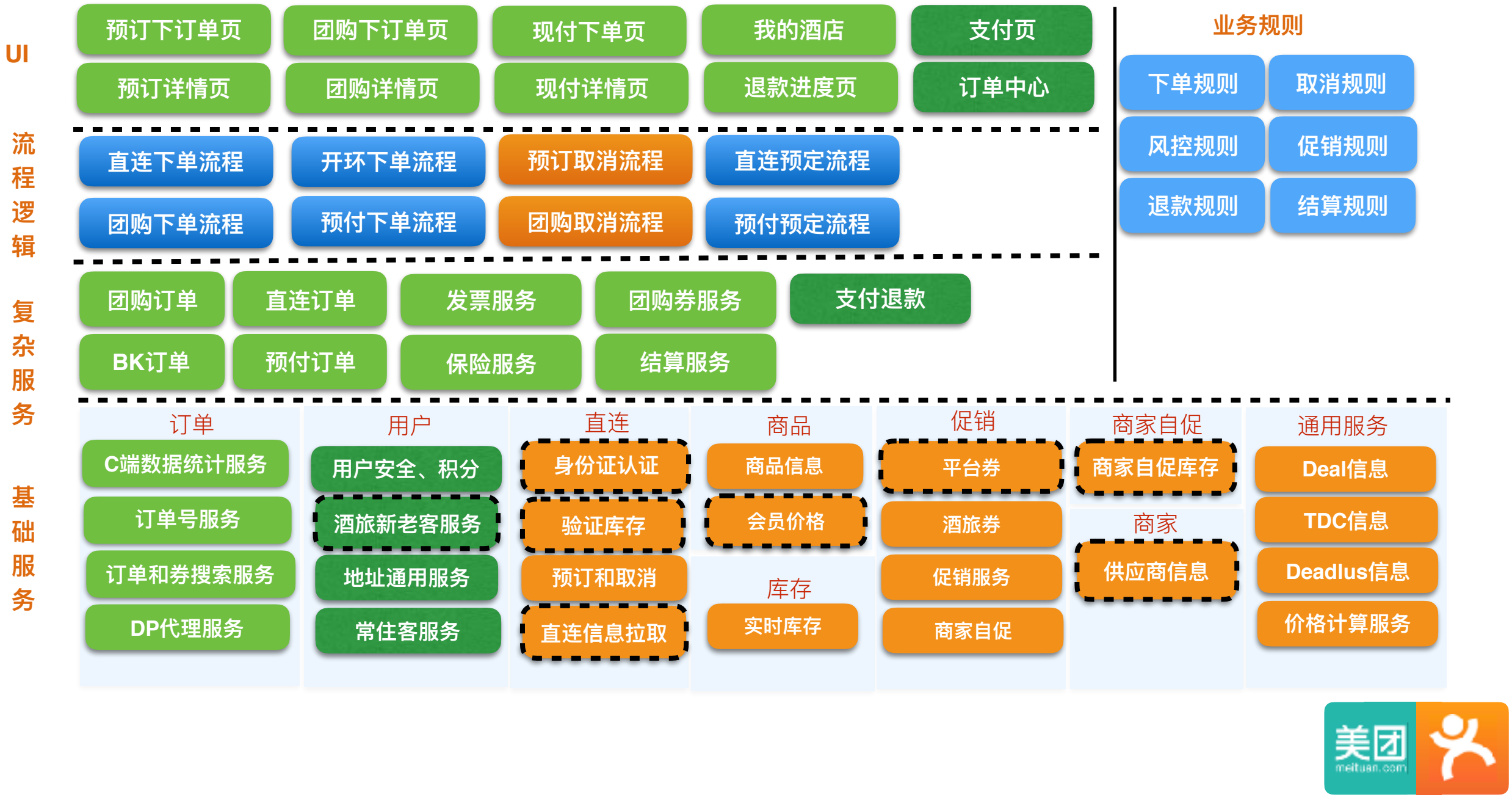
hotel-order+



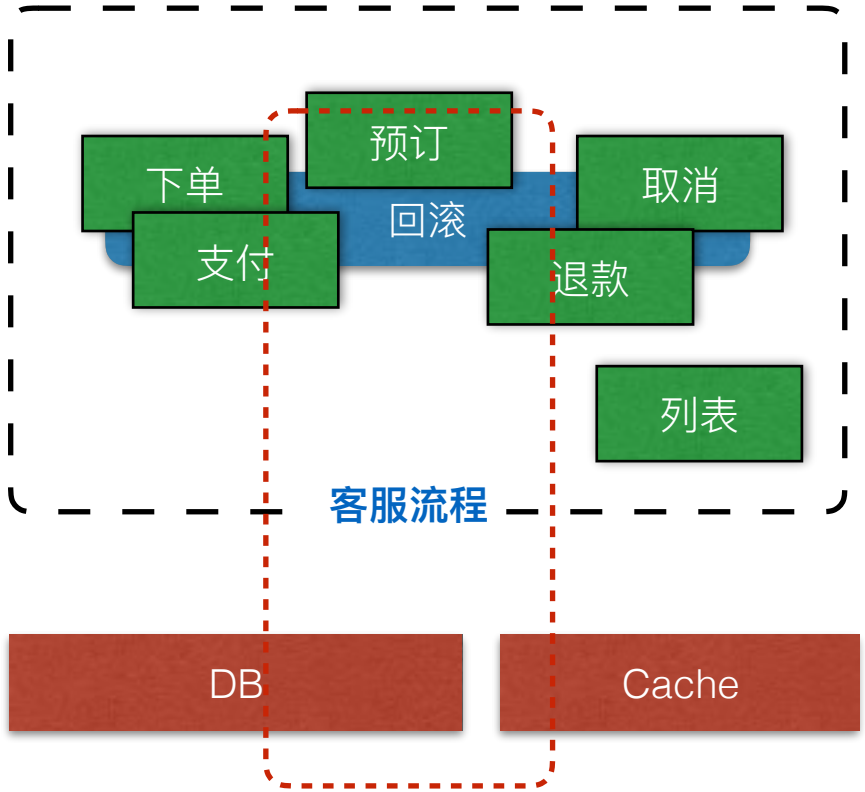
2016年6月

hotel-order+X

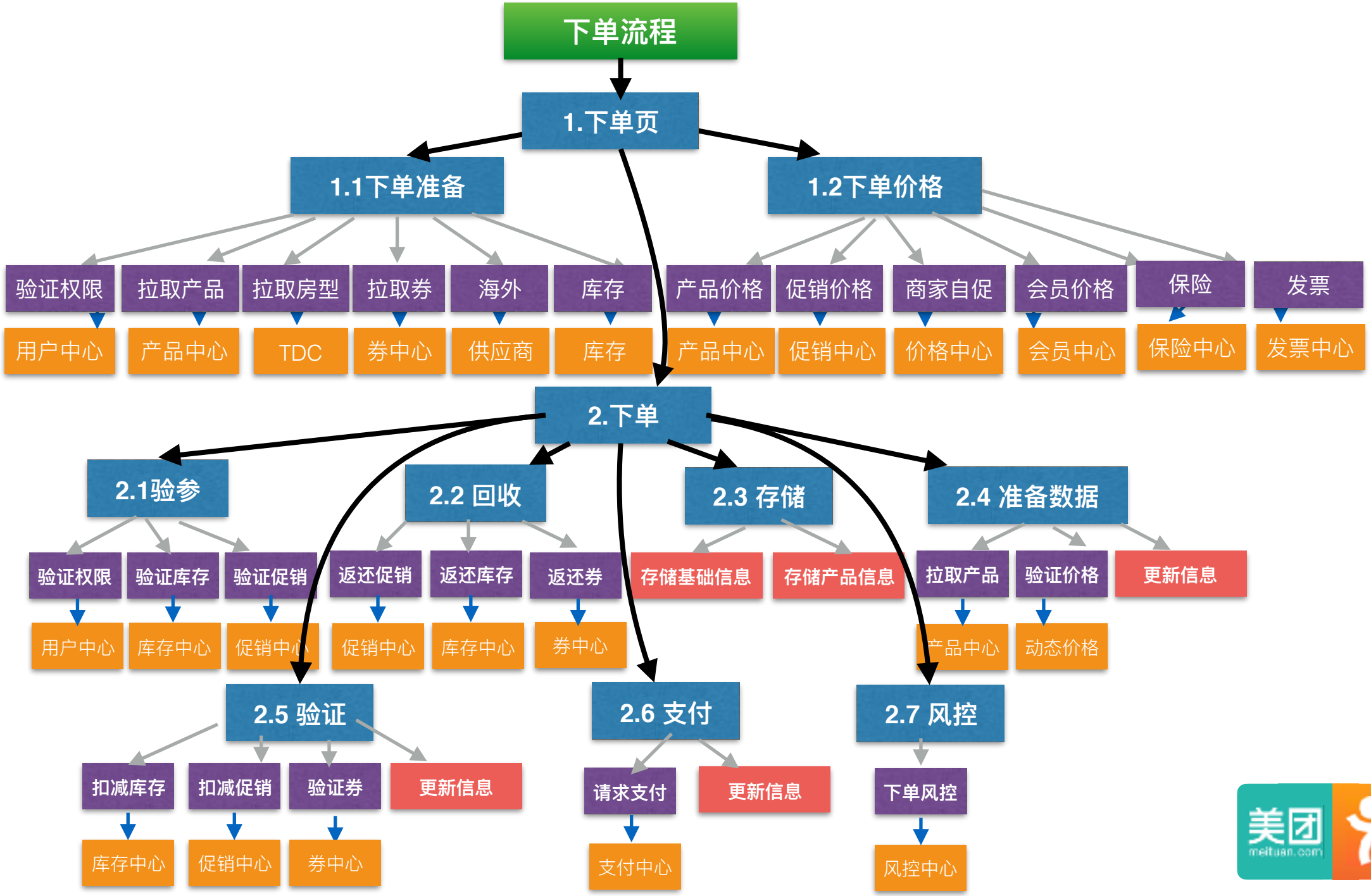



UI
流程逻辑
复杂服务
基础服务
预订下订单页
团购下订单页
现付下单页
我的酒店
支付页
预订详情页
团购详情页
现付详情页
退款进度页
订单中心
直连下单流程
开环下单流程
预订取消流程
直连预定流程
团购下单流程
预付下单流程
团购取消流程
预付预定流程
团购订单
直连订单
发票服务
团购券服务
支付退款
BK订单
预付订单
保险服务
结算服务
业务规则
下单规则
取消规则
风控规则
促销规则
退款规则
结算规则
订单
C端数据统计服务
订单号服务
订单和券搜索服务
DP代理服务
用户
用户安全、积分
酒旅新老客服务
地址通用服务
常住客服务
直连
身份证认证
验证库存
预订和取消
直连信息拉取
商品
商品信息
会员价格
库存
实时库存
促销
平台券
酒旅券
促销服务
商家自促
商家自促库存
商家
供应商信息
通用服务
Deal信息
TDC信息
Deadlus信息
价格计算服务

架构挑战-业务开发与系统重构



- 关键流程复杂交织，服务调用关系混乱
- 逻辑繁杂，降低了系统稳定性和系统质量
- 流程调用、数据存储糅杂
- 22+模块依赖使用


下单流程
1.下单页
1.1下单准备
1.2下单价格
验证权限
拉取产品
拉取房型
拉取券
海外
库存
产品价格
促销价格
商家自促
会员价格
保险
发票
用户中心
产品中心
TDC
券中心
供应商
库存
产品中心
促销中心
价格中心
会员中心
保险中心
发票中心
2.下单
2.1验参
2.2 回收
2.3 存储
2.4 准备数据
验证权限
验证库存
验证促销
返还促销
返还库存
返还券
存储基础信息
存储产品信息
拉取产品
验证价格
更新信息
用户中心
库存中心
促销中心
促销中心
库存中心
券中心
产品中心
动态价格
2.5 验证
2.6 支付
2.7 风控
扣减库存
扣减促销
验证券
更新信息
请求支付
更新信息
下单风控
库存中心
促销中心
券中心
支付中心
风控中心

架构挑战-稳定性与扩展性

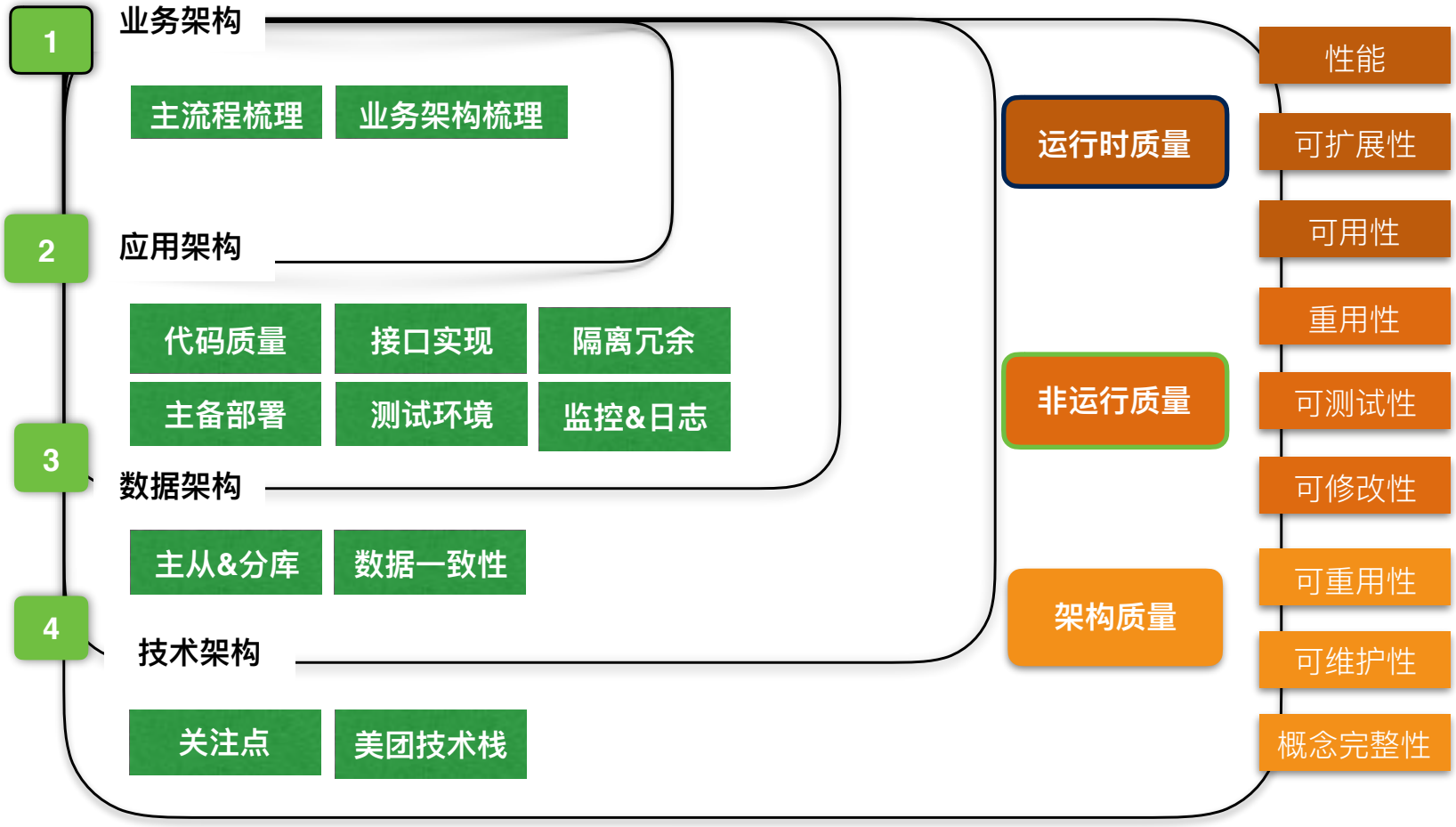
稳定性

- 可用性99.99%，核心系统99.999%（未来）。
- 设计20倍，实现5倍，发布3倍
- TP50 < 200ms,TP99 < 500ms

扩展性

- 松耦合
- 解析/拆分
- 抽象、幂等与补偿
- 容错设计
- 提高人员效率（周）

架构实践-关注点


1
业务架构
主流程梳理
业务架构梳理
2
应用架构
代码质量
接口实现
隔离冗余
主备部署
测试环境
监控&日志
3
数据架构
主从&分库
数据一致性
4
技术架构
关注点
美团技术栈
运行时质量
非运行质量
架构质量
性能
可扩展性
可用性
重用性
可测试性
可修改性
可重用性
可维护性
概念完整性

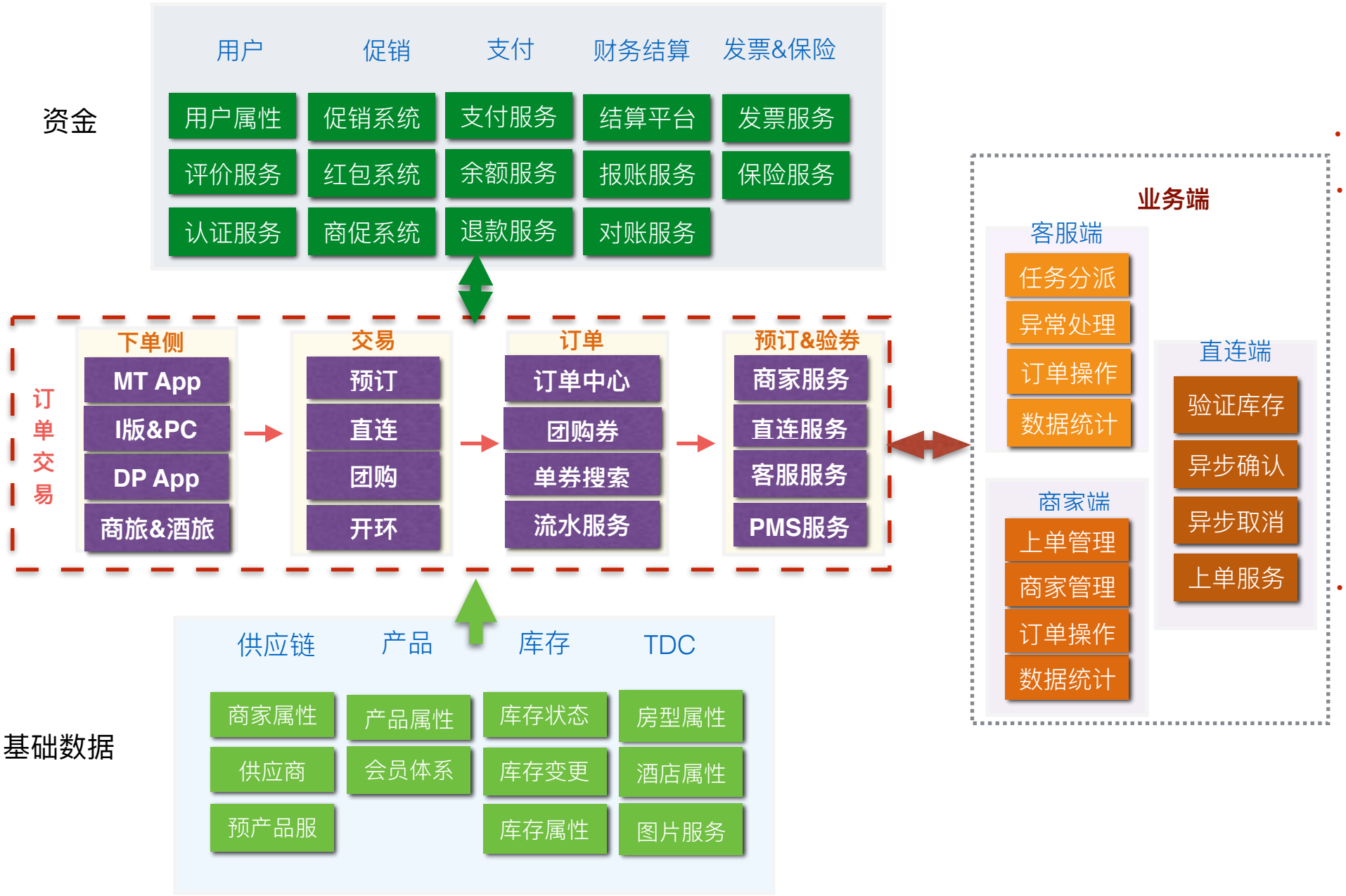
1 **业务架构** 主流程梳理 业务架构梳理

原则

- 核心与非核心业务分离
- 主流程与辅流程分离
- 不同业务隔离

方案

- **业务梳理**：领域模型，最小闭环，描述核心业务。
- **明确职责**：业务目标，团队目标，团队成员达成共识。
- **组织升级**：根据业务，设计团队，建设团队，提升团队。
- **流程优化**：需求管理、发版管理、故障管理、问题管理、配置记录等关键流程。



- **统一多端交易接口**
- **主流程优先保证**
  - 下单流程
  - 付款流程
  - 预订流程
  - 取消流程
  - 验券流程
  - 退款流程
- **辅流程异步实现**
  - 发票流程
  - 保险流程

C端核心功能

下单 | 支付 | 取消 | 退款 | 详情 | 隐藏

券中心 | 积分中心 | 促销中心 | 会员体系 | 发票平台

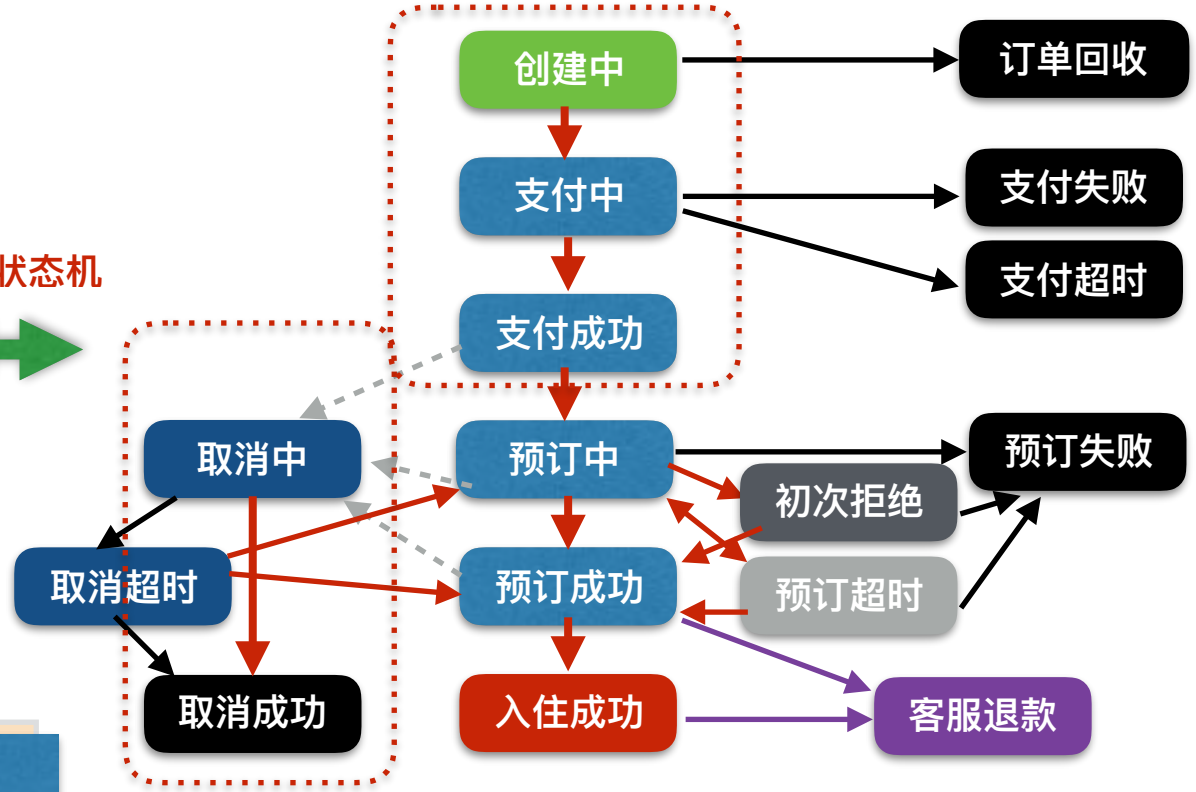
商家自促 | TDC | 保险平台

用户中心 | 库存中心 | 产品中心 | 供应商 | 价格服务

支付中心 | Deal中心 | Booking中心 | 直连服务 | 订单中心

关键依赖

订单状态机



B端核心功能

预付 | 确认预订 | 初次拒绝 | 确认拒绝

直连 | 预订确认 | 预订异常 | 取消确认 | 取消异常

团购 | 生成券 | 验券 | 重置券 | 冻结券

现付 | 接受订单 | 订单缓存

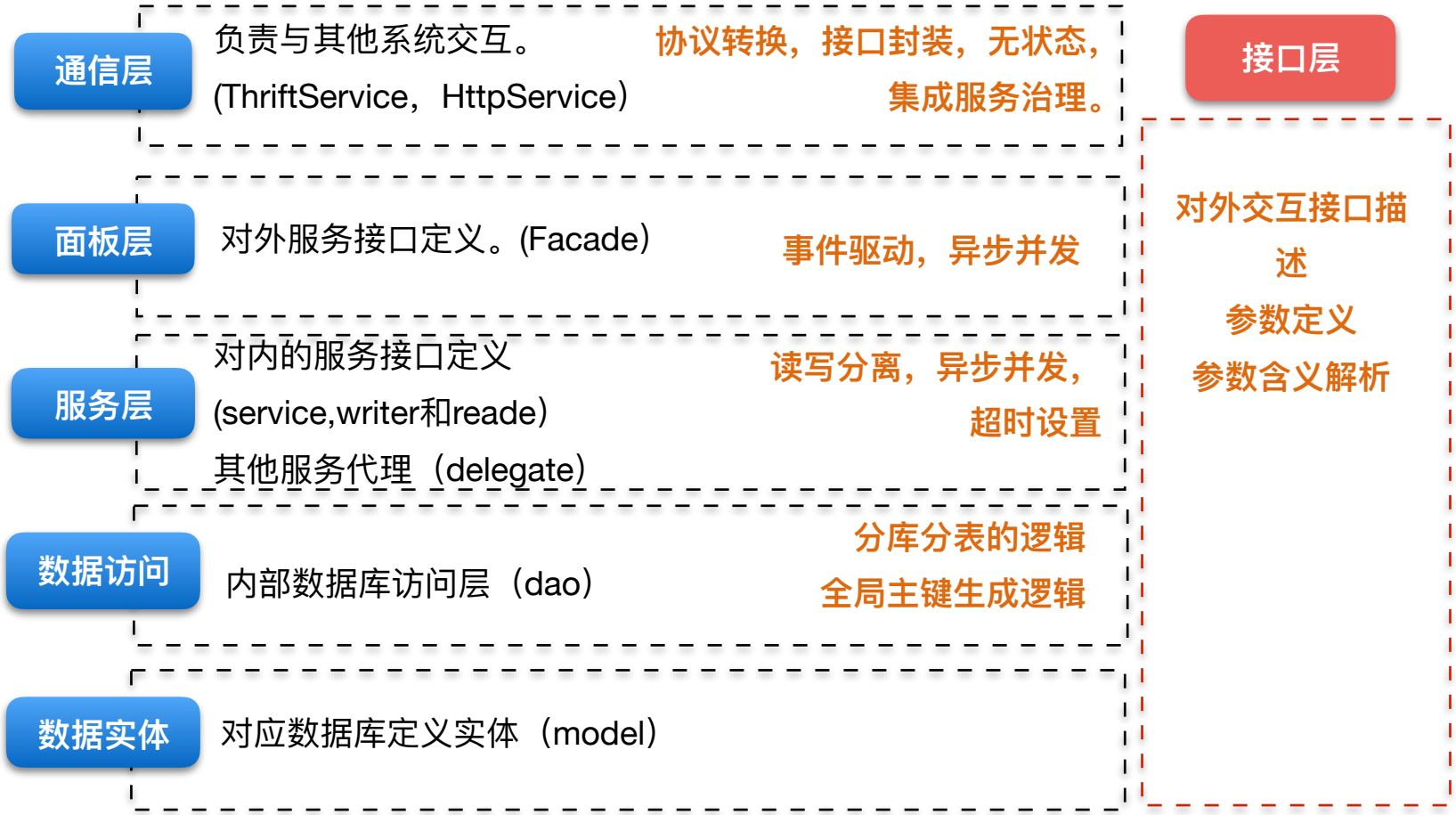
订单显示 | 复杂搜索 | 结算服务

- 核心业务精简，优先保证可用性（4个9），非核心多样化
- 隔离不同类型业务，非核心业务可以优先保证数据一致性

方案

- **开发视图：**
  - **代码质量**（模块化和分层，事件驱动开发）；
  - **质量保证**（测试环境，回归测试）。
- **发布视图：**
  - **灰度发布**（主倍，在线，回滚）；
  - **资源隔离**（硬件，数据，请求）；
  - **服务冗余**（服务，机房，数据）。
- **过程视图：**
  - **接口设计及实现**（幂等，补偿，读写分离，异步，并发，批量写入，职责单一）；
  - **服务治理**（在线扩容，安全保障，支持容错，故障转移）；
  - **服务降级**（超时，分功能降级，一键降级）；
  - **服务监控**（超时报警，流量监控）；
  - **性能压测**（线上压测，线下压测）；
  - **监控报警**（日志打包，业务监控，系统监控，基础硬件监控）

架构实践-开发视图-模块化和分层

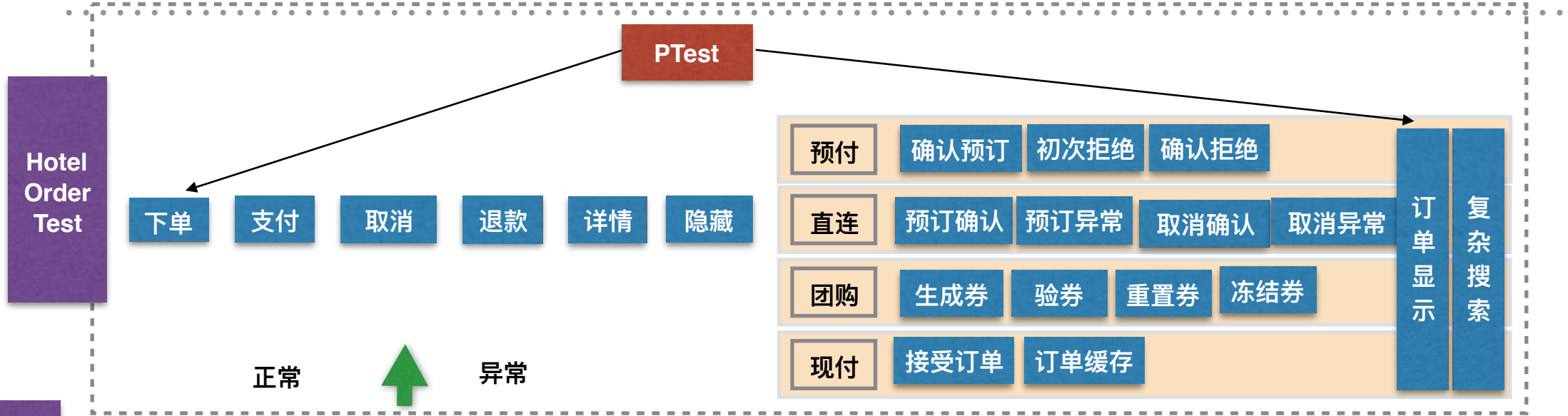


接口层

对外交互接口描述
参数定义
参数含义解析

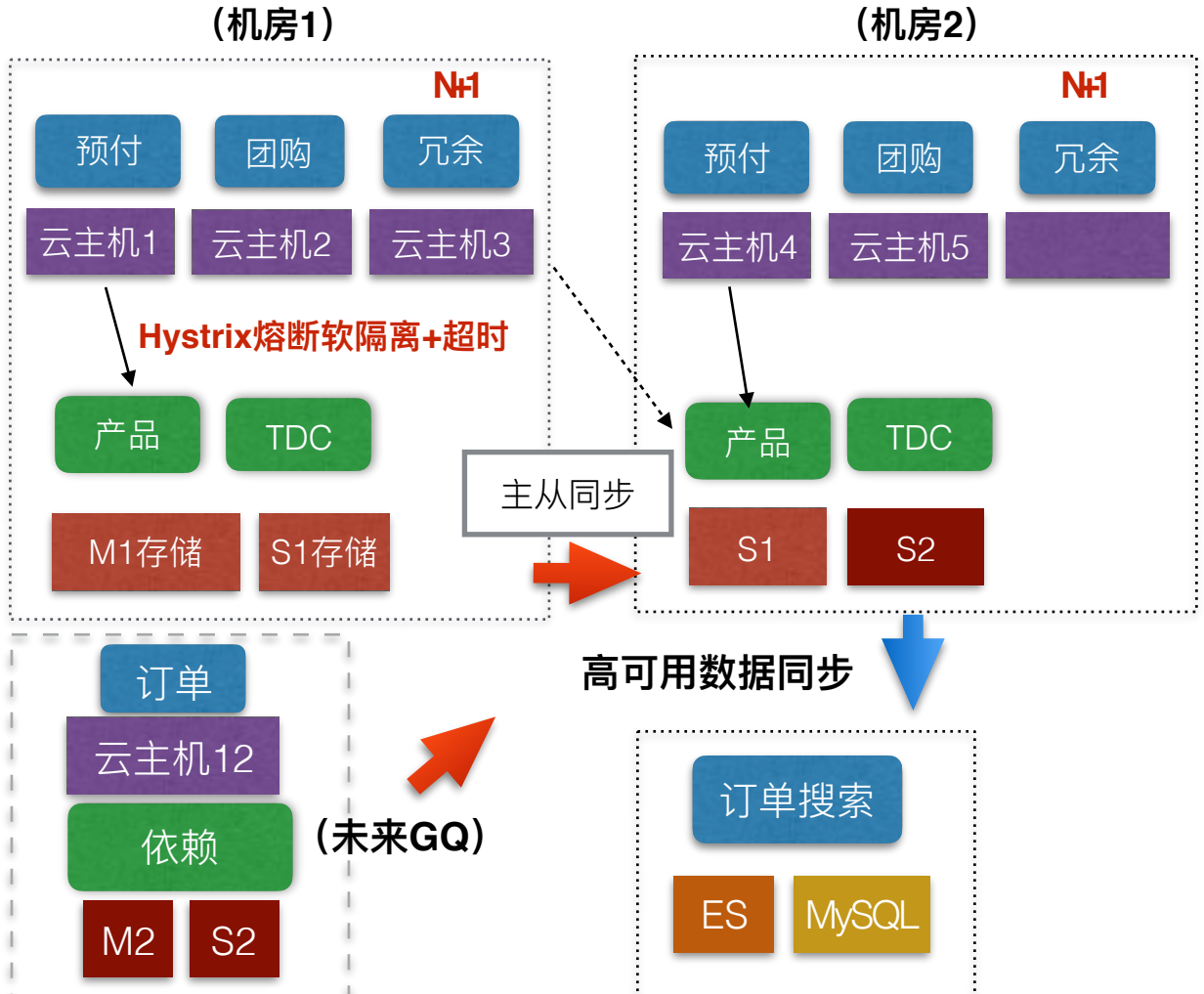
- **最严格：**
  - 编码要求
  - review要求
- **重视：**
  - 架构设计评审
  - 深入探讨设计中问题
- 集体代码走读，提高了团队对业

架构实践-开发视图-质量保证



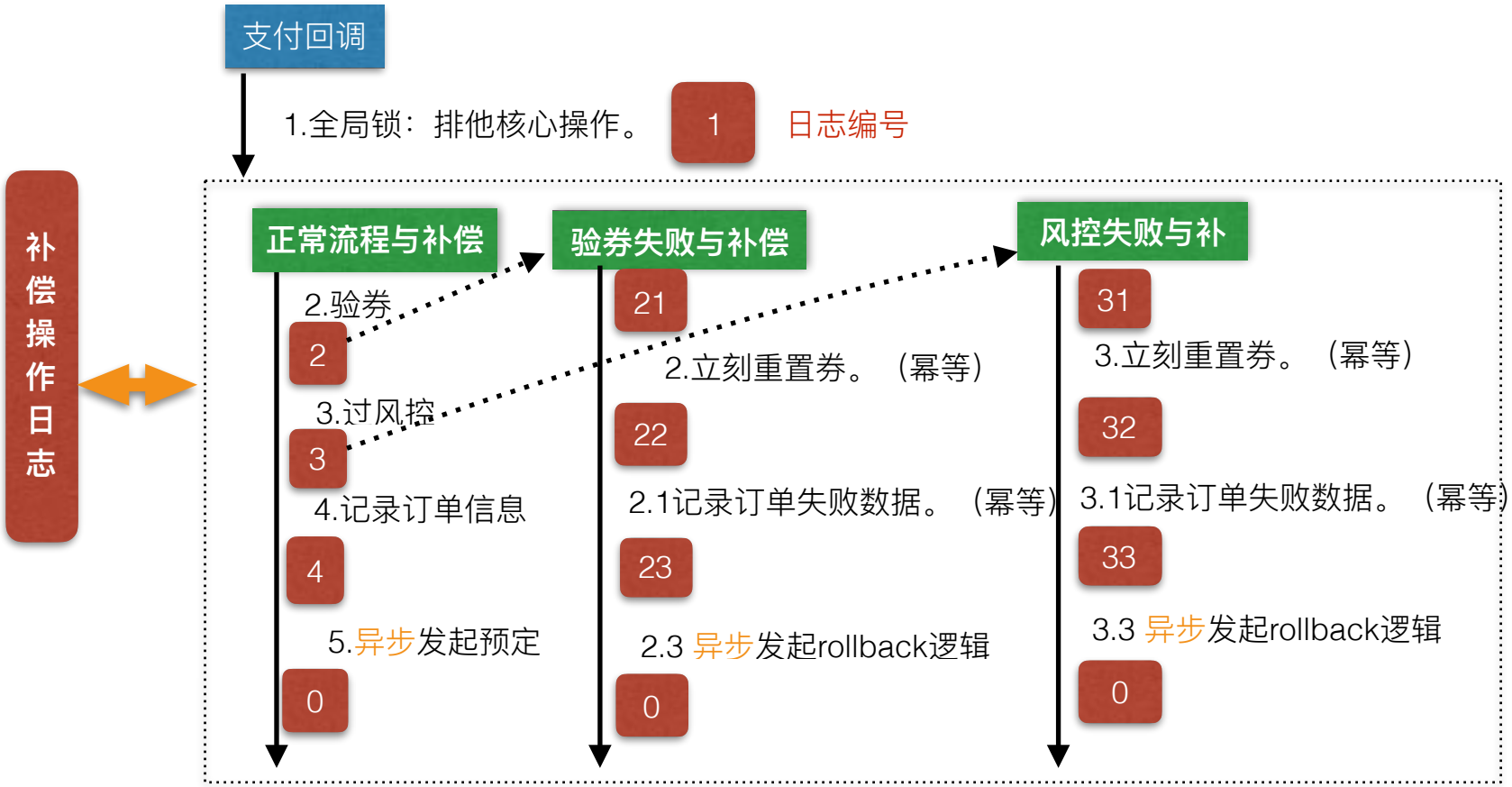
- 100%覆盖核心功能，完善回归和压力测试
- 多维度构建Mock系统，动态生产请求。（用户，商家，订单）
- 下单和搜索压测环境。

架构实践-发布视图-隔离&冗余



- 业务隔离：核心业务与非核心业务隔离
- 主机隔离：机房隔离，物理主机隔离，虚拟机隔离
- 应用隔离：服务分组隔离，依赖服务软隔离（熔断和超时）
- 应用冗余：3倍容量发布，N+1原则
- 存储冗余：主从存储。

架构实践-过程视图-接口幂等与补偿



- 补偿重试保证数据一致性
- 接口幂等，逻辑一致

15分钟时候发出报警和短信。1分钟内发报警邮件。　重试次数：1分钟,5分钟,10分钟,1小时,6小时,12小时,24小时。

架构实践-过程视图-服务治理

服务降级

- 服务降级：一键降级，分功能降级（接口）
- 过载保护：集群最大QPS，单机最大QPS，调用超时，异步，熔断

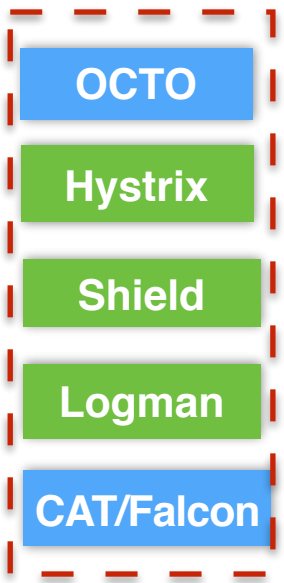
服务治理

- 在线扩容：服务发现注册
- 安全保障：黑白名单
- 故障转移：服务禁用
- 跨机房访问：服务转移

服务监控

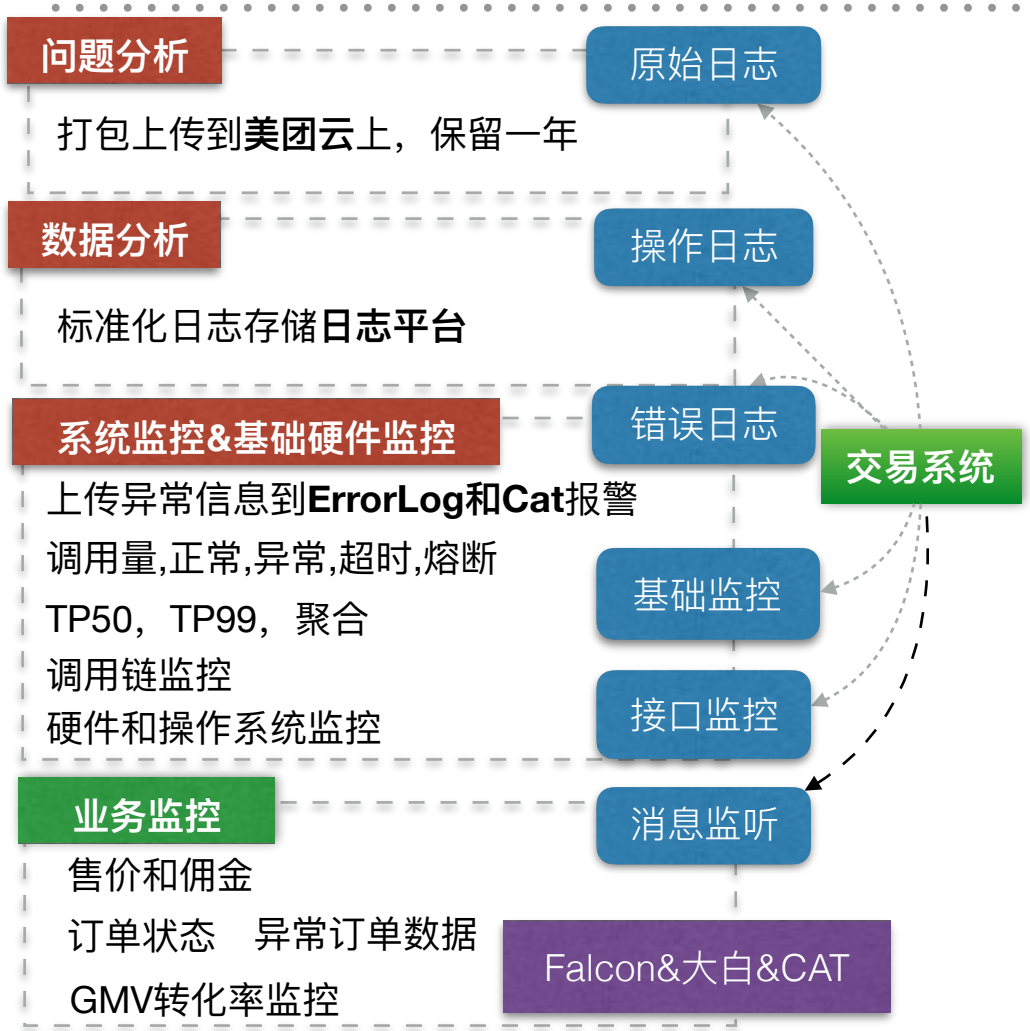
- 全链路追踪（覆盖接口、DB、KV、MQ、调度）
- 服务调用、超时、异常、错误报警（核心业务100%覆盖）
- 数据统计

扩展与自建



- 构建酒店交易服务治理平台
- 补充基础类库和服务

架构实践-过程视图-监控报警



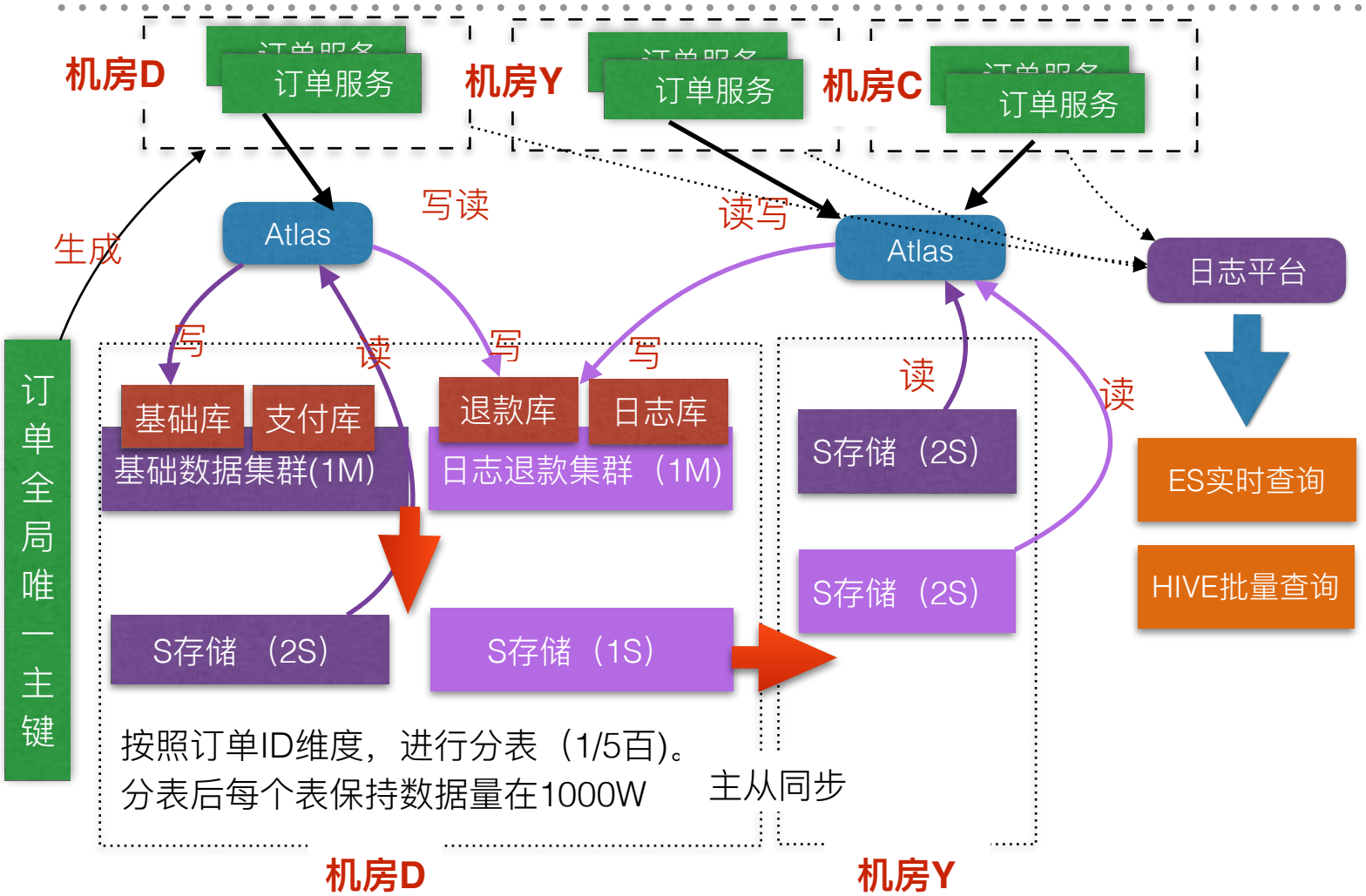
- 问题分析：证据保存，保存一年
- 数据分析：10m统计报告，1m找到原因
- 业务监控：秒级预警
- 系统监控&基础监控：分钟级别报警

3 **数据架构** 主从&分库 数据一致性

方案

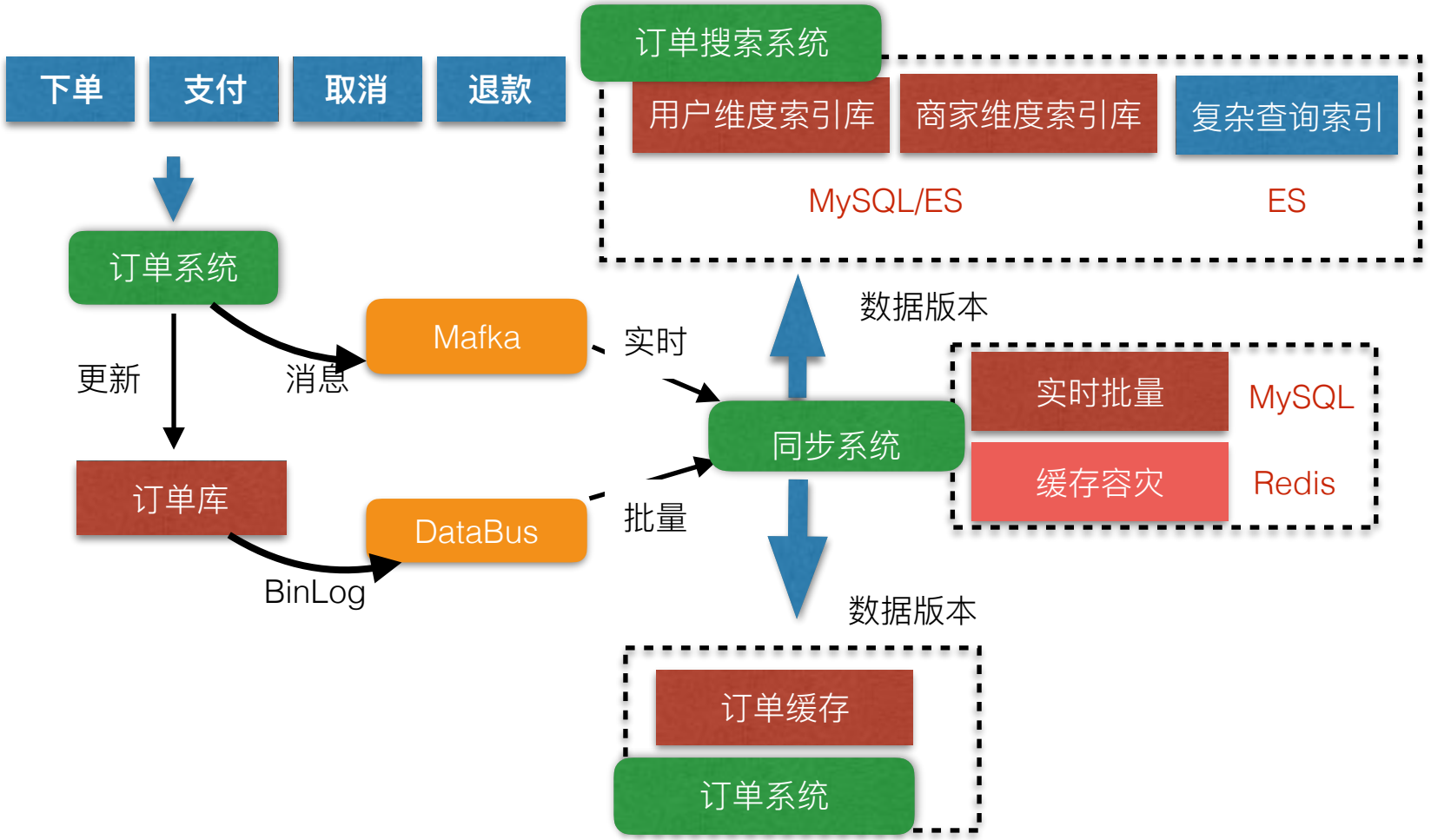
- **统一视图**：统一领域词语和数据库相关定义。及时性，一致性，准确性，完整性。
- **数据异构**：索引异构，数据库异构。
- **数据读写分离**：访问量大数据库做读写分离。存储量大数据库做分库分表。
- **数据最终一致**： 服务实现幂等和补偿，特定业务可以尝试TCC。线上Mysql，线下HIVE。合理使用缓存。

架构实践-数据架构-主从结构&分库分表



- 逻辑分库：业务场景划分,数据隔离
- 中间件：读写分离，分表逻辑
- 逻辑分库：业务场景划分,数据隔离
- 数据分片：全局自增主键划分，均匀分布
- 消除单点，保证可靠性，避免扩展瓶颈

架构实践-数据架构-数据异构与一致性

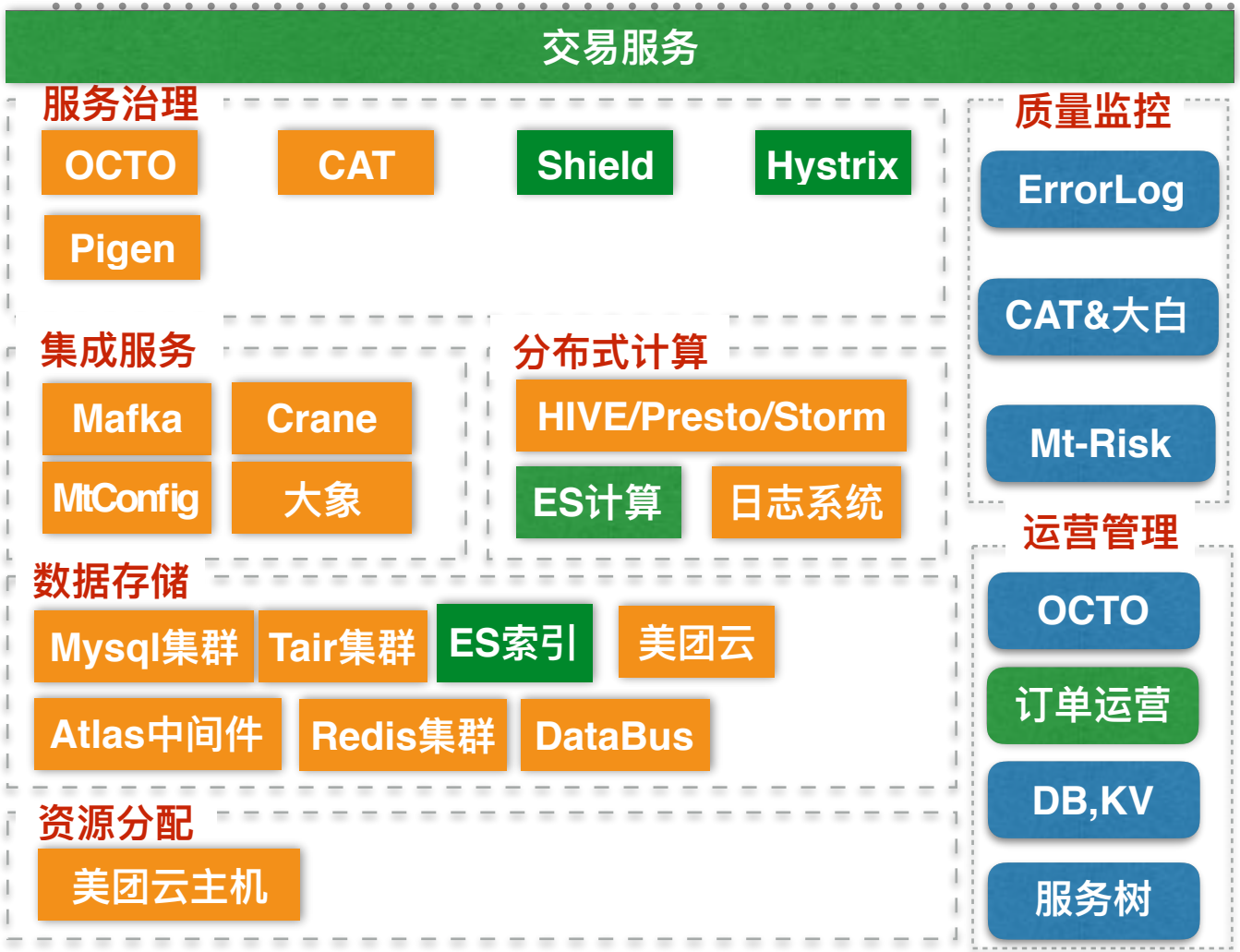


- 数据异构：搜索系统采用用户维度、商家维度时间维度划分索引，保证检索性能。
- 补偿机制：索引和缓存最终一致性
- 数据版本：保证了写入顺序，简化更新逻辑

方案

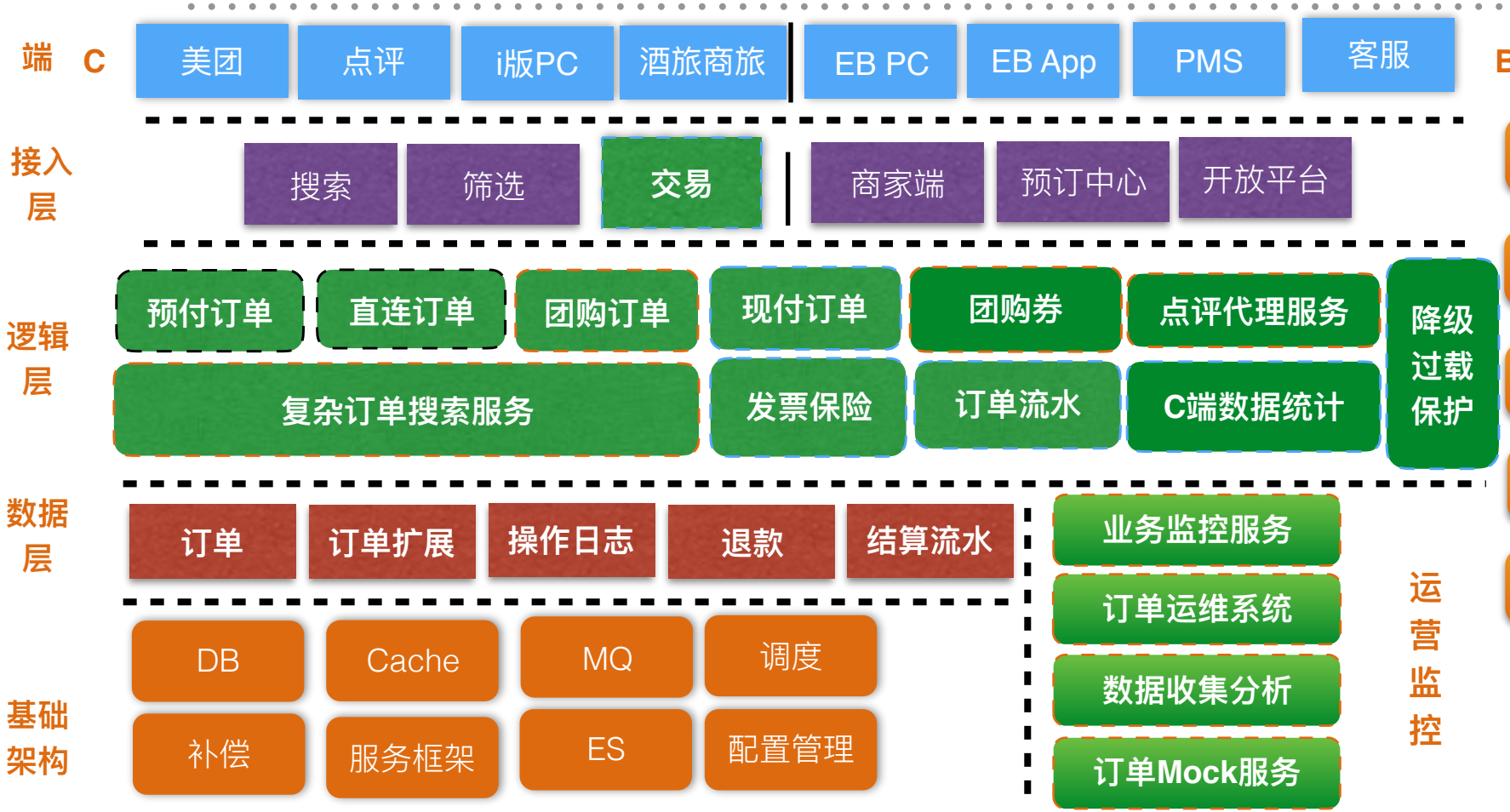
- **技术规划**：评估业务对技术需求，避免重复造轮子。
- **技术选型**：深入公司基础技术架构，升级底层基础类库。

架构实践-技术架构-交易技术栈



- 交易依赖技术栈构建对接支持
- 完善技术最佳实践方案，完成全员培训
- 补充技术栈：服务治理、计算、存储、运营

架构实践-成果-交易统一&微服务



15Q2

15Q4

16Q1

- 交易业务统一：
  业务研发统一，制定联调方案。
- 微服务化：
  拆分成16个子工程业务，保证核心业务稳定，并提高开发效率。

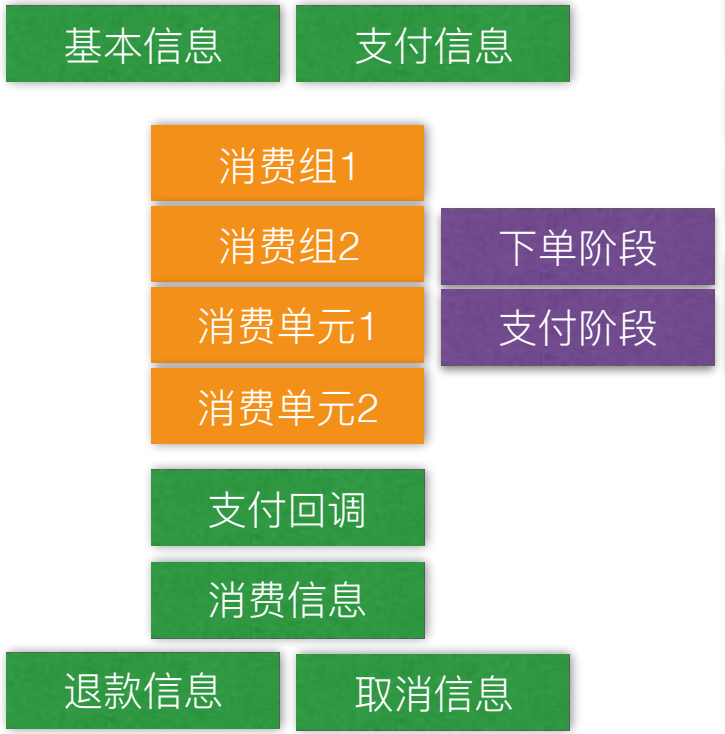
架构实践-成果-稳定性&运行时质量&非运行时质量

稳定性	0 宕机事故		99.99% 可用性		
运行时	5 倍 核心接口容量	2 倍 下单耗时优化	天->秒 数据问题发现	小时->分钟 问题定位	小时->分钟 问题修复
非运行时	月->周 新功能上线	线上->线下 功能测试	天->分钟 业务测试耗时	100% 核心业务测试	天->分钟 问题统计分析

未来规划-平台化

- 订单搜索：3个维度提供搜索服务
- 结算流水：业务流水逻辑固话
- 数据同步：保证数据最终一致性
- 订单服务：订单共性服务
- 订单业务：订单规则和状态机，消费与退款

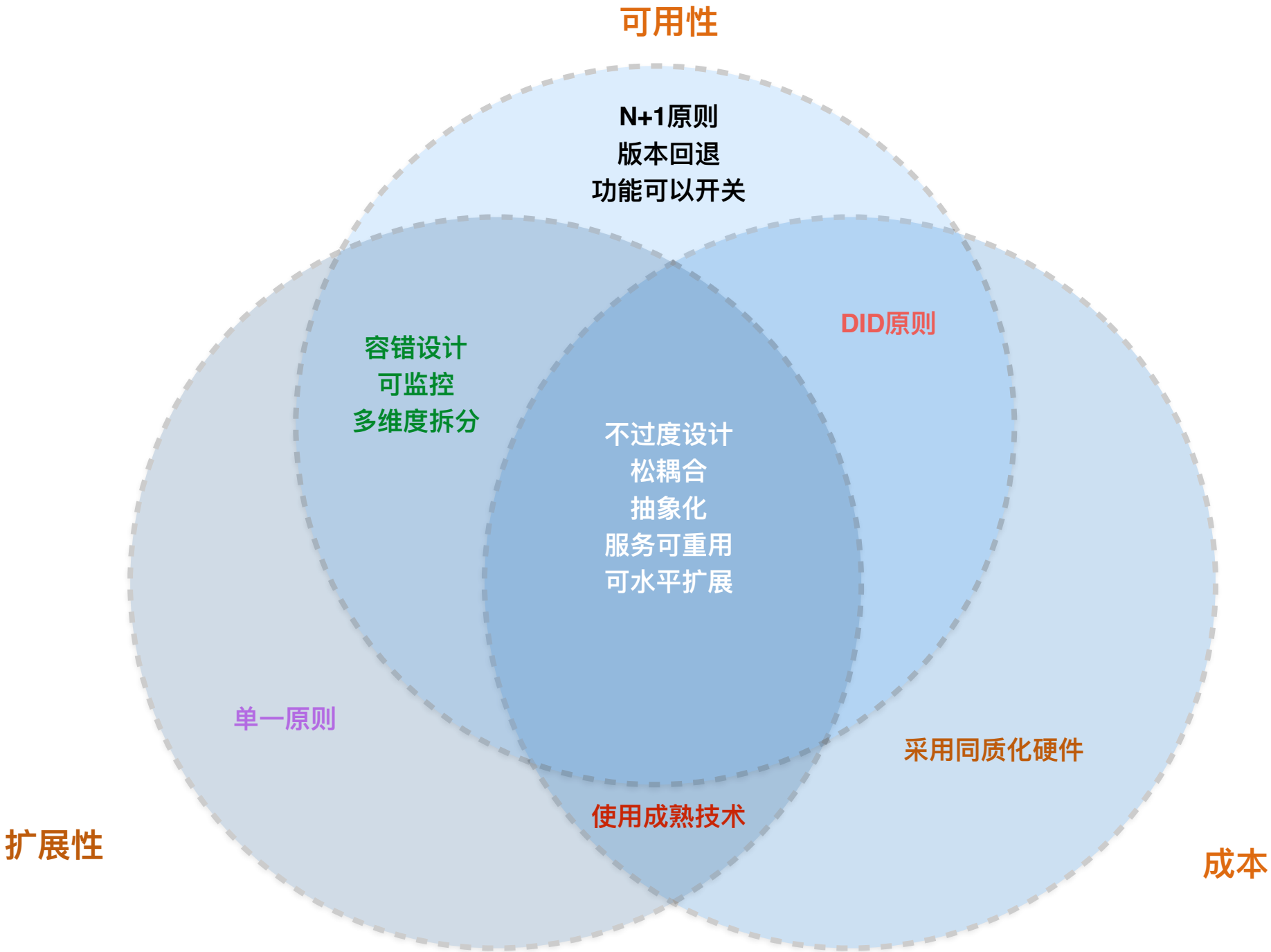
未来规划-平台化-模型抽象



基础信息	userId,parterId,订单状态，隐藏，评论，退款状态和消费状态， 业务自定义状态码，平台自定义状态码，C端展示ID，更新版本
阶段信息	订单ID，下单，支付等不同订单状态静态信息（收款定义，优惠定义，取消定义，退款定义），版本
消费组	消费组可以包含多个消费单元也可以包含多个消费组，但是不能同时都包含
消费单元	退款状态和消费状态,,订单ID，外部消费组或者单元ID，单元描述 消费单元(组）状态（例如团购：创建，使用，重置，撤销，冻结，取消）
下单：收款定义	收款类型，金额，金额单位，外部消费组或者单元ID 底价，佣金，佣金百分比
下单：优惠定义	优惠类型，金额，金额单位，外部消费组或者单元ID， 美团承担，商家承担，佣金百分比，优惠维度（订单维度），优惠ID
下单：取消定义	取消扣款类型，扣款逻辑，扣款版本，返还逻辑，返还版本
下单：退款定义	支持退款类型，逻辑，版本
支付信息	收用户总钱数。汇率，应收总钱数，优惠总钱数，美团承担，商家承担，支付平台必要信息

谢谢大家

QA


可用性
N+1原则
版本回退
功能可以开关
容错设计
可监控
多维度拆分
DID原则
不过度设计
松耦合
抽象化
服务可重用
可水平扩展
单一原则
采用同质化硬件
使用成熟技术
扩展性
成本